

## SUBQUERIES -- CHAPTER 6 --

- A Subquery is a SELECT statement that is embedded in a clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries. They can be very useful when trying to select rows from a table with a condition that depends on the data in the table itself.
- You can place the subquery in a number of SQL clauses:
  - WHERE clause
  - HAVING clause
  - FROM clause
- Here's the syntax:

```
SELECT    select_list
FROM      table
WHERE     expr operator
          (SELECT    select_list
           FROM      table);
```

- Note that comparison operators fall under one of two classes: single-row operators (>, =, >=, <, <>, <=) and multiple-row operators (IN, ANY, ALL).
- Note that a subquery must appear on the right of a comparison operator and must be enclosed within a pair of parentheses.
- There's different types of subqueries:
- Single-row queries return one row from the inner SELECT statement and uses a 'single-row' operator (mentioned earlier).
  - **Single-row** subquery --> CLERK

```
SQL> SELECT ENAME, JOB
       2 FROM EMP
       3 WHERE JOB =
       4           (SELECT JOB
       5              FROM EMP
       6              WHERE EMPNO = 7369);
```

ENAME	JOB
SMITH	ANALYST
SCOTT	ANALYST
ADAMS	ANALYST
H	ANALYST
"HELLO"	ANALYST
FREY	ANALYST
...	

- Here's an example using the HAVING clause with a single-row subquery...

```
SQL> SELECT deptno, min(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING min(sal) >
5 (SELECT min(sal)
6 FROM emp
7 WHERE deptno = 20);
```

```
DEPTNO  MIN(SAL)
-----  -
          40          5000
```

- As shown here, and as mentioned earlier, you can have subqueries in the WHERE clause as well as in the HAVING clause.

- **Multiple-row** subquery --> CLERK  
MANAGER

- These queries return more than one row. For these, you must use the multiple-row comparison operators which are IN, ANY, ALL, as mentioned earlier.

```
SQL> SELECT ENAME, SAL, DEPTNO
2 FROM EMP
3 WHERE SAL IN (SELECT MIN(SAL)
4 FROM EMP
5 GROUP BY DEPTNO);
```

```
ENAME          SAL      DEPTNO
-----  -
HAYES          99        50
ALLEN         1000       30
WARD          1000       30
TURNER        1000       30
HAYES2        1000       50
...
```

- **Multiple-column** subquery --> CLERK 7900  
MANAGER 7698

- Using the ANY operator in Multiple-Row subqueries...The ANY operator (and its synonym SOME operator) compares a value to each value returned by a subquery. See the following example...

```
SQL> SELECT EMPNO, ENAME, JOB
```

```

2 FROM EMP
3 WHERE SAL < ANY
4             (SELECT SAL
5              FROM EMP
6              WHERE JOB = 'CLERK')
7 AND JOB <> 'CLERK';

```

no rows selected

- Although no rows were returned, the query is well formed.
- Finally let's look at how the ALL operator is used in Multiple-Row Subqueries...

```

SQL> SELECT EMPNO, ENAME, JOB
2 FROM EMP
3 WHERE SAL > ALL
4             (SELECT AVG(SAL)
5              FROM EMP
6              GROUP BY DEPTNO);

```

EMPNO	ENAME	JOB
9666	SHANK	ANALYST

- The ALL operator compares the value to every value returned by a subquery.
- NOTE that the NOT operator may be used in conjunction with any of IN, ANY, ALL operators.